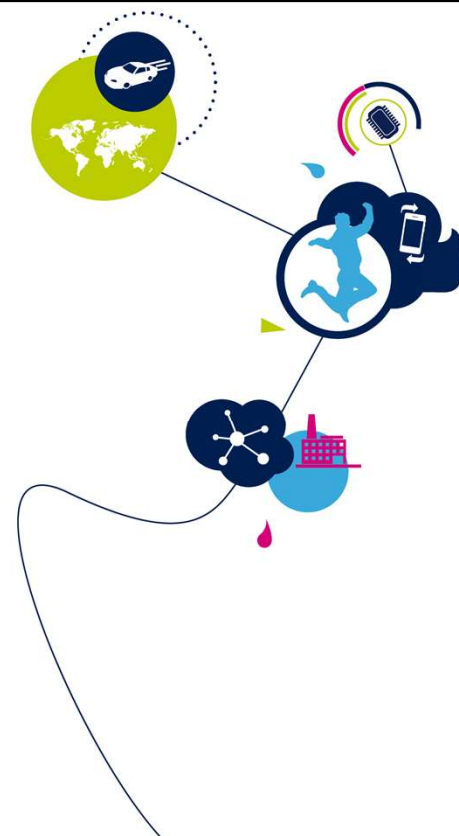


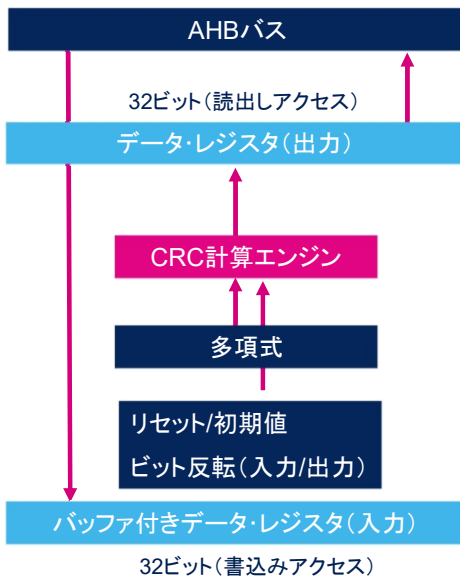
STM32G4 - CRC

巡回冗長検査計算ユニット

1.0版



STM32巡回冗長検査計算ユニットのプレゼンテーションによろそ。ここでは、エラー検出に使用されているコードを生成するCRCユニットの機能のすべてをカバーしています。



- 8、16、32ビットデータからのCRCコード生成に使用
 - 完全に設定可能な奇数多項式
 - プログラム可能なサイズ
 - 入出力データは完全可逆

アプリケーション側の利点

- データ整合性を検証
- ソフトウェア・コード・シグネチャを生成
- CPUとDMAのどちらでも簡単に使用可能

CRCとは、巡回冗長検査のことです。CRC計算ユニットは、設定可能な多項式値とサイズを用いて8、16、32ビットデータのCRCコードを生成するために使用されます。CRCに基づくテクニックは、データ転送やストレージの整合性を確認するために使用できます。

CRC計算は、実行時のアプリケーションソフトウェアのシグネチャ計算にも使用できます。リンク時に生成されて、特定のメモリ領域に保存されたリファレンスシグネチャと計算されたソフトウェアシグネチャが比較されます。

STM32のCRC計算ユニットは完全に設定可能ですので、ソフトウェアのオーバーヘッドは最小限に抑えられます。さらに、DMAコントローラは、CPUが別のタスクを実行しているか、SLEEPモードにある間に、大きなデータブロックに対する連続CRC計算のために使用できます。

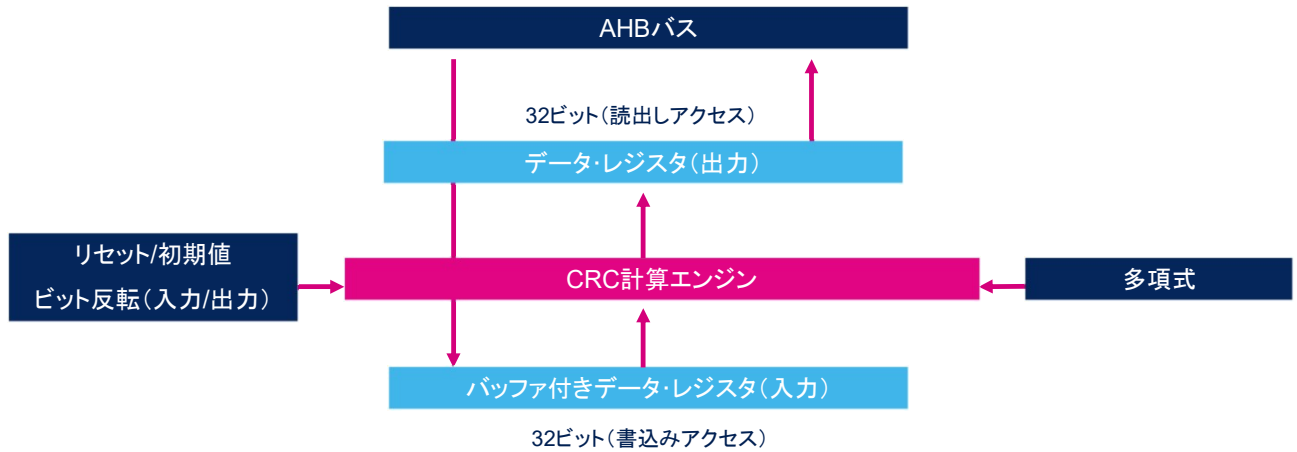
- 完全にプログラム可能な奇数多項式:
 - デフォルトでは、CRC-32 (Ethernet) 多項式 (0x04C11DB7) を使用
 - これは、 $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$ という多項式を表す
- プログラム可能な多項式サイズ (7、8、16、32ビット)
- プログラム可能な初期値 (デフォルトは0xFFFFFFFF)
- 入出力データ用各種エンディアン・スキーム



STM32に搭載されているCRC計算ユニットは、サイズが7、8、16、32ビットにプログラム可能で、完全にプログラム可能な奇数多項式を特徴としています。

初期値も完全にプログラム可能であり、非常に柔軟な実行時のCRCコード生成が可能となっています。

ペリフェラルはビッグエンディアンまたはリトルエンディアン形式の入出力データに対応するように設定可能であり、各種の通信プロトコルに対応しています。



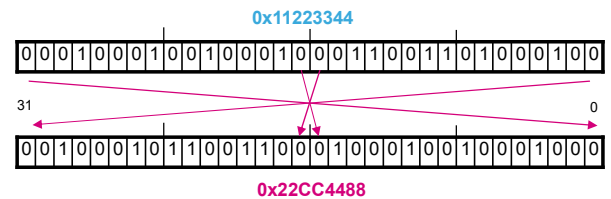
CRC計算ユニットは、最適な性能を出すためにAHBバスインタフェースに接続されています。

32ビットレジスタが1個含まれており、CPUとDMAコントローラのいずれかによる書込みと読出しの両方のために使用されます。

初期値、入出力ビット反転、および、多項式係数は、新たにCRC計算を始める前の初期化フェーズ中に、ソフトウェアによって設定されます。CRC計算エンジンは過去のCRC計算結果をすべて保持しており、これが次の計算の初期値となりますので、データブロックの新たなCRC計算が可能となります。

ソフトウェア・オーバーヘッドを削減

- 入力データは、8、16、32ビット単位で反転可能
 - CRC_CRレジスタのREV_IN[1:0]値に依存
 - たとえば、入力データが0x1A2B3C4Dの場合、CRC計算では以下のように使用される：
 - 0x58D43CB2(バイト単位でビットが反転)
 - 0xD458B23C(ハーフワード単位でビットが反転)
 - 0xB23CD458(フルワードでビットが反転)
- 出力データは32ビット単位で反転可能
 - CRC_CRレジスタのREV_OUTビットを設定
 - 出力データ0x11223344の例を右図に示す



入出力フォーマットは、ソフトウェアオーバーヘッドの削減のために完全に設定可能です。そのため、アプリケーションは各種エンディアンスキームの管理が可能です。

CRC_CRレジスタのREV_IN[1:0]ビットによって、8、16、32ビット単位で入力データの反転と実行が可能です。

たとえば、入力データが0x1A2B3C4Dの場合、CRC計算では以下のように使用されます。:

- 0x58D43CB2(バイト単位でビットが反転)
- 0xD458B23C(ハーフワード単位でビットが反転)
- 0xB23CD458(フルワードでビットが反転)

また、出力データも、CRC_CRレジスタのREV_OUTビットをセットすることによって反転することができます。

操作はビットレベルで行われます。たとえば、出力データ0x11223344は0x22CC4488に変換されます。

- 単一の入出力32ビットデータレジスタによってファームウェア操作が簡単
- AHBバスがストールしない連続書込み用入力バッファ
- CRC計算時間は入力データ幅に依存:
 - 32ビットブロックの場合、4AHBクロック・サイクル
 - 16ビットブロックの場合、2AHBクロック・サイクル
 - 8ビットブロックの場合、1AHBクロック・サイクル
- 書込みはCPUがSLEEPモードの間にDMAにより実行可能
 - 動作は低消費電力(連続計算)で、CPUは時間要求の厳しいタスクのために開放



life.augmented

CRCデータレジスタには入力バッファが含まれており、実行中のCRC計算によるウェイトステートを待つことなく、CPUまたはDMAペリフェラルによる2番目のデータの書込みが直ちに可能です。CRCデータレジスタは、ワード、右詰めのハーフワード、右詰めのバイトによってアクセスできます。

計算の時間はデータ幅に依存します:

- 32ビットブロックの場合、4AHBクロックサイクル
- 16ビットブロックの場合、2AHBクロックサイクル
- 8ビットブロックの場合、1AHBクロックサイクル

DMAコントローラはCRCエンジンへの書込みに使用できます。これによって、CPUが別のタスクのためにオフロードされ、コードまたはデータの整合性の連続計算に用いることが可能となります。

モード	説明
RUN	有効
低電力RUN	有効
SLEEP	有効 DMAコントローラはCRC計算に使用可能
低電力SLEEP	有効 DMAコントローラはCRC計算に使用可能
STOP	STOP0、STOP1、STOP2モードで停止 ペリフェラル・レジスタの内容は保たれる
STANDBY	パワーダウン状態 CRCユニットは、STANDBYモード終了後に再初期化する必要がある
SHUTDOWN	パワーダウン状態 CRCユニットは、SHUTDOWNモード終了後に再初期化する必要がある



このスライドには、CRC計算ユニットを使用可能な電源モードの概要が示されています。

CRC計算ユニットは、動作にメインAHBクロックが必要です。